Hepatoprotective effects of systemic ER activation

Human ER-sensitive genes and NAFLD classification models

Christian Sommerauer & Carlos Gallardo

25 July, 2023

```r
# source and library import
source('code/00_helper_functions.R')
library(tidyverse)
library(RColorBrewer)
library(ComplexHeatmap)
library(caret)
library(multiROC)
library(patchwork)
```

```r
# color palettes
colPals <- list()
colPals$conditions <- setNames(c('#44AA99', '#117733', '#88CCEE', '#332288', '#DDCC77', '#CC6677', '#AA4499', '#882255'),
                               c('CDf', 'HFDf', 'CDm', 'HFDm', 'DPN', 'DIP', 'E2', 'PPT'))
colPals$RdBu <- rev(RColorBrewer::brewer.pal(n=11, name = 'RdBu'))
colPals$UpDown <- setNames(colPals$RdBu[c(10,2)],
                           c('up', 'down'))
colPals$clusters <- setNames(c('#E6E6E6', '#B3B3B3', '#8C8C8C', '#4D4D4D'),
                             c('1', '2', '3', '4'))
colPals$celltypes <- setNames(c('#B4272F', '#E5462D', '#FFD1D1', '#F4E54C', '#FBAA3E', '#AA654E', '#B58B80', '#6D7AA5',
                                '#B3177E', '#CAC1DD', '#67227D','#36B449', '#82C349', '#A9D265', '#199478', '#95A3A3', '#C4C5C7'),
                              c('Cholangiocytes', 'Endothelial cells', 'HsPCs', 'Stromal cells', 'Hepatocytes',
                                'Kupffer cells', 'Monocytes & Monocyte-derived cells', 'T cells', 'NK cells',
                                'ILC1s', 'B cells', 'cDC1s', 'cDC2s', 'Mig. cDCs', 'pDCs', 'Basophils', 'Neutrophils'))
colPals$inferno <- c('#FCFFA4', '#FCA50A', '#DD513A', '#932667', '#420A68', '#000004')
colPals$models <- setNames(c('#F6B651', '#92A3A5', '#972D22'),
                           c('stage', 'nas', 'fibrosis'))
colPals$models_light <- setNames(c('#FAD7A0', '#BFC9CA', '#DF7B71'),
                                 c('stage', 'nas', 'fibrosis'))
```

Load data

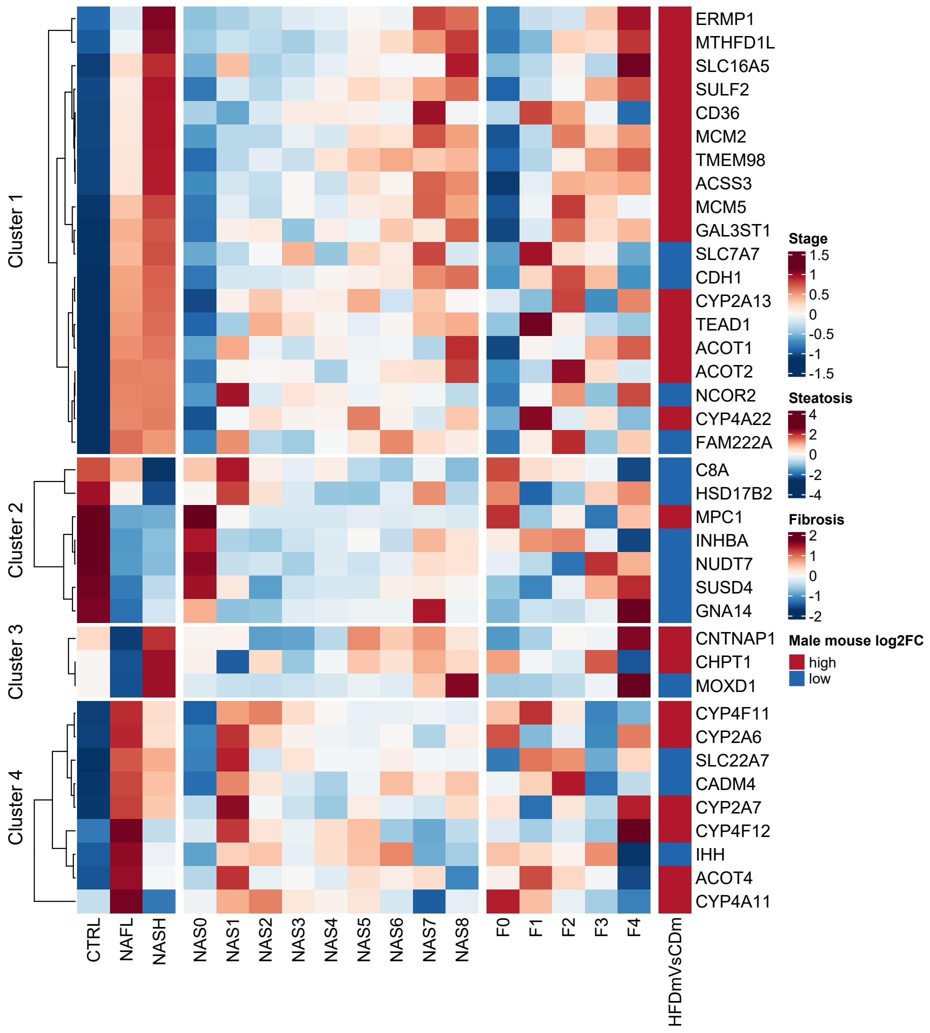
```r
# mouse estrogen-sensitive enhancer-gene pairs
mouse_45_ER_genes <- read.table(
  file = 'results/Epigenome_analysis/corr_45genes_67enh_toPlot.txt',
  stringsAsFactors = FALSE,
  sep = '\t',
  header = TRUE) %>%
  dplyr::pull(symbol) %>%
  unique()

# mouse-human orthologs
mouse_human_orthologs <- read.table(
  file = 'data/ensembl_mmus_hsap_dec2021_orthologs.tsv',
  sep = '\t',
  header = TRUE,
  quote = '')

# mouse differentially expressed genes
DEGs <- readRDS('results/bulkRNAseq_mmus_DEGs.rds')

# NAFLD patient cohort
cohort_data <- readRDS("data/bulkRNAseq_human_cohort_data.rds")

cohort_data$Govaere$cpm_filt <- cohort_data$Govaere$cpm %>%
  tibble::rownames_to_column(var = 'gene') %>%
  dplyr::filter(gene %in% mouse_human_orthologs$GeneID_human) %>%
  dplyr::mutate(gene = dplyr::recode(gene, !!!setNames(mouse_human_orthologs$GeneSymbol_human,
                                                        mouse_human_orthologs$GeneID_human))) %>%
  dplyr::filter(!duplicated(gene) & gene != "") %>%
  tibble::column_to_rownames(var = 'gene')
```

Convert estrogen-sensitive genes in mouse to human orthologs

```r
# find genes that are expressed in human patients. Cutoff: 0.5 CPM.
human_detectable <- cohort_data$Govaere$cpm_filt %>%
  apply(., 1, median)
human_detectable <- names(human_detectable[human_detectable > 0.5])

# filter ER genes for those expressed in human
human_ER_genes <- mouse_human_orthologs %>%
  dplyr::filter(GeneSymbol_mouse %in% mouse_45_ER_genes) %>%
  dplyr::filter(GeneSymbol_human %in% human_detectable) %>%
  dplyr::filter(!duplicated(GeneSymbol_human)) %>%
  dplyr::pull(GeneSymbol_human)

# export human ER genes
cat(human_ER_genes,
    file = 'results/ER_sensitive_genes.txt',
    sep = '\n')
```

Integrated heatmap of ER-sensitive genes across human NAFLD spectrum

```r
human_stage <- cohort_data$Govaere$cpm_filt %>%
  dplyr::filter(rownames(.) %in% human_ER_genes) %>%
  groupTransform(cohort_data$Govaere$meta$Stage, function(x) apply(x,1,median)) %>%
  scaleData(method = 'zscore') %>%
  dplyr::arrange(rownames(.))

human_steatosis <- cohort_data$Govaere$cpm_filt %>%
  dplyr::filter(rownames(.) %in% human_ER_genes) %>%
  groupTransform(cohort_data$Govaere$meta$NAS, function(x) apply(x,1,median)) %>%
  scaleData(method = 'zscore') %>%
  dplyr::arrange(rownames(.))

human_fibrosis <- cohort_data$Govaere$cpm_filt %>%
  dplyr::filter(rownames(.) %in% human_ER_genes) %>%
  groupTransform(cohort_data$Govaere$meta$Fibrosis, function(x) apply(x,1,median)) %>%
  scaleData(method = 'zscore') %>%
  dplyr::arrange(rownames(.))

mouse_log2FC_HFDmVsCDm <- DEGs$filt$CDmVsHFDm %>%
  dplyr::rename(GeneSymbol_mouse = external_gene_name) %>%
  dplyr::inner_join(mouse_human_orthologs, by = 'GeneSymbol_mouse') %>%
  dplyr::filter(GeneSymbol_human %in% human_ER_genes) %>%
  dplyr::filter(!duplicated(GeneSymbol_human)) %>%
  dplyr::select(GeneSymbol_human, log2FoldChange) %>%
  dplyr::mutate(log2FoldChange = log2FoldChange*-1) %>%
  dplyr::rename(HFDmVsCDm = log2FoldChange) %>%
  dplyr::mutate(HFDmVsCDm = factor(ifelse(HFDmVsCDm>0, 'high', 'low'), levels = c('high','low'))) %>%
  tibble::column_to_rownames(var = 'GeneSymbol_human') %>%
  dplyr::arrange(rownames(.))

set.seed(4)
clusters_ER_genes <- kmeans(human_stage, centers = 4, iter.max = 100)

Heatmap(human_stage, width = unit(24, "mm"),name = "Stage",
        split = clusters_ER_genes$cluster,
        col = circlize::colorRamp2(breaks=seq(-max(abs(human_stage)), max(abs(human_stage)), length.out=11),
                                   colors=colPals$RdBu),
        row_title = c("Cluster 1", "Cluster 2", "Cluster 3", "Cluster 4"),
        cluster_row_slices = FALSE,
        column_order=c("CTRL", "NAFL", "NASH")) +
  Heatmap(human_steatosis, width = unit(72, "mm"),name = "Steatosis",
          column_order=paste0('NAS', seq(0,8)),
          col = circlize::colorRamp2(breaks=seq(-max(abs(human_steatosis)), max(abs(human_steatosis)), length.out=11),
                                     colors=colPals$RdBu)) +
  Heatmap(human_fibrosis, width = unit(40, "mm"),name = "Fibrosis",
          column_order=paste0('F', seq(0,4)),
          col = circlize::colorRamp2(breaks=seq(-max(abs(human_fibrosis)), max(abs(human_fibrosis)), length.out=11),
                                   colors=colPals$RdBu)) +
  Heatmap(mouse_log2FC_HFDmVsCDm, width = unit(8, "mm"), name = "Male mouse log2FC",
          col = colPals$RdBu[c(10,2)])
```



NAFLD classification models

NAFLD models

```
models <- list()
models[['Stage']] <- buildModel(dat = t(cohort_data$Govaere$cpm_filt),
                                group = factor(cohort_data$Govaere$meta$Stage, levels = c('CTRL','NAFL','NASH')),
                                method = 'glmnet',
                                features = human_ER_genes,
                                preproc = c('scale', 'center'),
                                train_test_split = 0.6,
                                tune_iter = 10,
                                seed = 22)

models[['NAS']] <- buildModel(dat = t(cohort_data$Govaere$cpm_filt),
                              group = factor(cohort_data$Govaere$meta$NAS_class, levels = c('low','mid','high')),
                              method = 'glmnet',
                              features = human_ER_genes,
                              preproc = c('scale', 'center'),
                              train_test_split = 0.6,
                              tune_iter = 10,
                              seed = 22)
```

```r
models[['Fibrosis']] <- buildModel(dat = t(cohort_data$Govaere$cpm_filt),
                                   group = factor(cohort_data$Govaere$meta$Fibrosis_class, levels = c('low','mid','high')),
                                   method = 'glmnet',
                                   features = human_ER_genes,
                                   preproc = c('scale', 'center'),
                                   train_test_split = 0.6,
                                   tune_iter = 10,
                                   seed = 22)

saveRDS(models, file = 'results/nafld_classification_models.rds')
```

Random gene set models

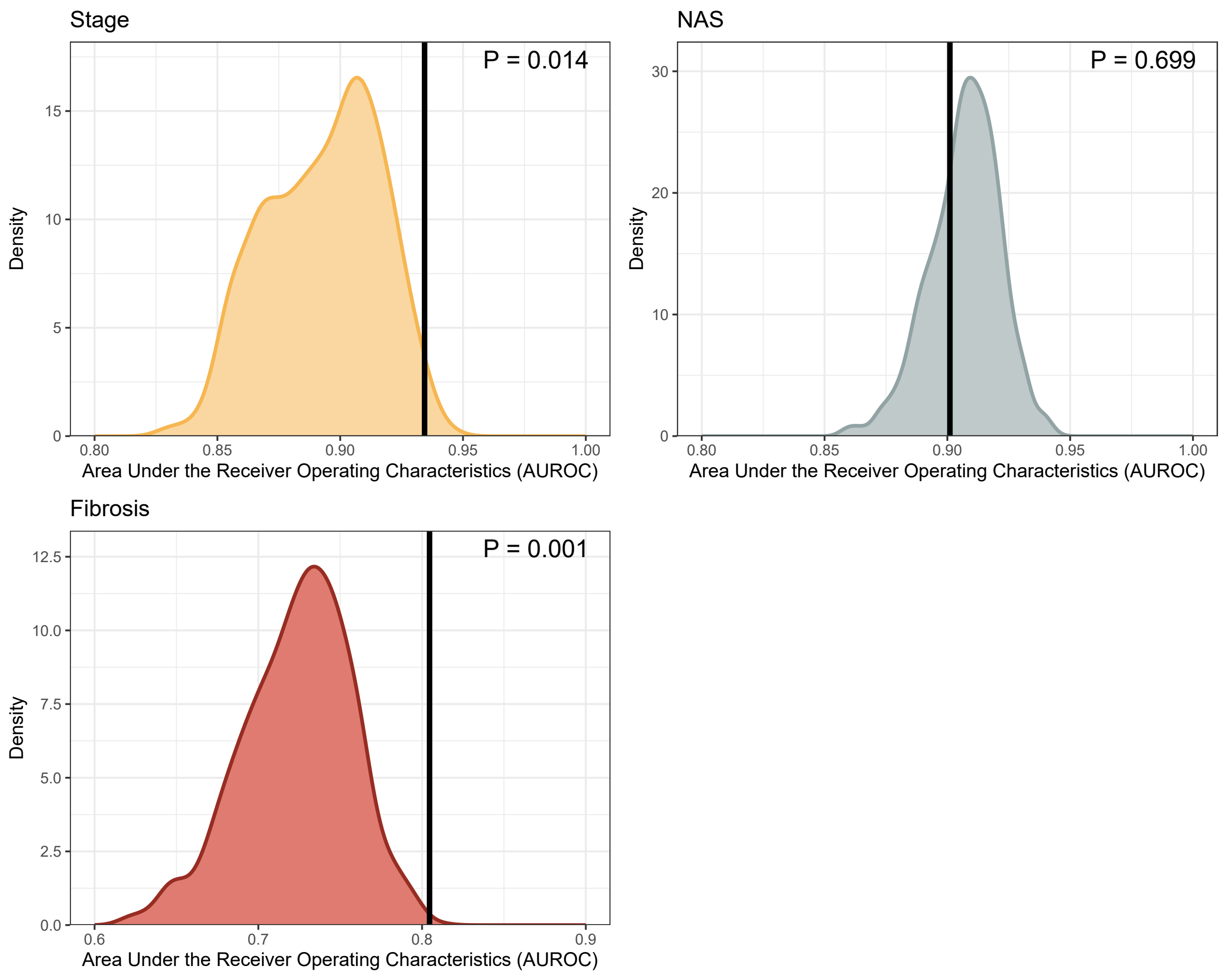
```r
# random gene sets for null distribution of prediction models
# all_genes <- rownames(cohort_data$Govaere$cpm_filt)
# random_genes <- lapply(seq(1,1000), function(x) sample(x = all_genes, size = length(human_ER_genes)))
# saveRDS(random_genes, file = 'results/random_38gene_subsets.rds')

# load random gene sets for reproducibility
random_genes <- readRDS(file = 'results/random_38gene_subsets.rds')
```

```r
# train random models (very long computing time!)
# random_models <- list()
# random_models[['Stage']] <- lapply(random_genes, function(x) {
#   buildModel(dat = t(cohort_data$Govaere$cpm_filt),
#              group = factor(cohort_data$Govaere$meta$Stage, levels = c('CTRL','NAFL','NASH')),
#              method = 'glmnet',
#              features = x,
#              preproc = c('scale', 'center'),
#              train_test_split = 0.6,
#              tune_iter = 10,
#              seed = 22)
# })
# random_models[['NAS']] <- lapply(random_genes, function(x) {
#   buildModel(dat = t(cohort_data$Govaere$cpm_filt),
#              group = factor(cohort_data$Govaere$meta$NAS_class, levels = c('low','mid','high')),
#              method = 'glmnet',
#              features = x,
#              preproc = c('scale', 'center'),
#              train_test_split = 0.6,
#              tune_iter = 10,
#              seed = 22)
# })
# random_models[['Fibrosis']] <- lapply(random_genes, function(x) {
#   buildModel(dat = t(cohort_data$Govaere$cpm_filt),
#              group = factor(cohort_data$Govaere$meta$Fibrosis_class, levels = c('low','mid','high')),
#              method = 'glmnet',
#              features = x,
#              preproc = c('scale', 'center'),
#              train_test_split = 0.6,
#              tune_iter = 10,
#              seed = 22)
# })

# saveRDS(random_models, file = 'Results/random_models_38genes.rds')

random_models <- readRDS(file = 'results/random_models_38genes.rds')
```

Predictive potential of ER regulated genes vs random gene sets

```r
p <- list()

df <- data.frame(auc = lapply(random_models$Stage, function(x) x$resultsROC$AUC$model$micro) %>% unlist())
val <- models$Stage$resultsROC$AUC$model$micro
pval <- tailFraction(val, df$auc, tail = 'right')
p[[1]] <- ggplot(df, aes(x = auc)) +
  geom_density(lwd = 1, colour = '#F6B651',
               fill = '#FAD7A0', alpha = 1) +
  geom_vline(xintercept = val, lwd = 1.5) +
  scale_x_continuous(limits = c(0.8,1)) +
  scale_y_continuous(expand = expansion(mult = c(0, .1))) +
  annotate("text",  x=Inf, y = Inf, label = paste('P =',pval), vjust=1.5, hjust=1.2, size=5) +
  xlab('Area Under the Receiver Operating Characteristics (AUROC)') +
  ylab('Density') +
  ggtitle('Stage') +
  theme_bw()
```

```r
df <- data.frame(auc = lapply(random_models$NAS, function(x) x$resultsROC$AUC$model$micro) %>% unlist())
val <- models$NAS$resultsROC$AUC$model$micro
pval <- tailFraction(val, df$auc, tail = 'right')
p[[2]] <- ggplot(df, aes(x = auc)) +
  geom_density(lwd = 1, colour = '#92A3A5',
               fill = '#BFC9CA', alpha = 1) +
  geom_vline(xintercept = val, lwd = 1.5) +
  scale_x_continuous(limits = c(0.8,1)) +
  scale_y_continuous(expand = expansion(mult = c(0, .1))) +
  annotate("text",  x=Inf, y = Inf, label = paste('P =',pval), vjust=1.5, hjust=1.2, size=5) +
  xlab('Area Under the Receiver Operating Characteristics (AUROC)') +
  ylab('Density') +
  ggtitle('NAS') +
  theme_bw()

df <- data.frame(auc = lapply(random_models$Fibrosis, function(x) x$resultsROC$AUC$model$micro) %>% unlist())
val <- models$Fibrosis$resultsROC$AUC$model$micro
pval <- tailFraction(val, df$auc, tail = 'right')
p[[3]] <- ggplot(df, aes(x = auc)) +
  geom_density(lwd = 1, colour = '#972D22',
               fill = '#DF7B71', alpha = 1) +
  geom_vline(xintercept = val, lwd = 1.5, color = 'black') +
  scale_x_continuous(limits = c(0.6,0.9)) +
  scale_y_continuous(expand = expansion(mult = c(0, .1))) +
  annotate("text",  x=Inf, y = Inf, label = paste('P =',pval), vjust=1.5, hjust=1.2, size=5) +
  xlab('Area Under the Receiver Operating Characteristics (AUROC)') +
  ylab('Density') +
  ggtitle('Fibrosis') +
  theme_bw()

patchwork::wrap_plots(p, nrow=2, ncol=2, byrow=T)
```



SessionInfo

```r
sessionInfo()
```

```
## R version 4.2.1 (2022-06-23 ucrt)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
## Running under: Windows 10 x64 (build 19045)
##
## Matrix products: default
##
## locale:
## [1] LC_COLLATE=English_United States.utf8
## [2] LC_CTYPE=English_United States.utf8
## [3] LC_MONETARY=English_United States.utf8
## [4] LC_NUMERIC=C
## [5] LC_TIME=English_United States.utf8
##
## attached base packages:
## [1] grid      stats     graphics  grDevices utils     datasets  methods
## [8] base
##
## other attached packages:
##  [1] patchwork_1.1.2       multiROC_1.1.1        caret_6.0-92
##  [4] lattice_0.20-41       ComplexHeatmap_2.14.0 RColorBrewer_1.1-3
##  [7] lubridate_1.9.2       forcats_1.0.0         stringr_1.5.0
## [10] dplyr_1.1.0           purrr_1.0.1           readr_2.1.4
## [13] tidyr_1.3.0           tibble_3.2.1          ggplot2_3.4.2
## [16] tidyverse_2.0.0
##
## loaded via a namespace (and not attached):
##  [1] nlme_3.1-162         matrixStats_0.63.0   doParallel_1.0.17
##  [4] tools_4.2.1          utf8_1.2.2           R6_2.5.1
##  [7] rpart_4.1.19         BiocGenerics_0.44.0  colorspace_2.0-3
## [10] nnet_7.3-19          GetoptLong_1.0.5     withr_2.5.0
## [13] tidyselect_1.2.0     compiler_4.2.1       glmnet_4.1-7
## [16] cli_3.4.1            Cairo_1.6-0          labeling_0.4.2
## [19] scales_1.2.1         proxy_0.4-27         digest_0.6.30
## [22] rmarkdown_2.23       pkgconfig_2.0.3      htmltools_0.5.5
## [25] parallelly_1.36.0    highr_0.10           fastmap_1.1.0
## [28] rlang_1.1.1          GlobalOptions_0.1.2  rstudioapi_0.15.0
## [31] farver_2.1.1         shape_1.4.6          generics_0.1.3
## [34] zoo_1.8-12           ModelMetrics_1.2.2.2 magrittr_2.0.3
## [37] Matrix_1.5-3         Rcpp_1.0.9           munsell_0.5.0
## [40] S4Vectors_0.36.0     fansi_1.0.3          lifecycle_1.0.3
## [43] stringi_1.7.8        pROC_1.18.4          yaml_2.3.7
## [46] MASS_7.3-60          plyr_1.8.8           recipes_1.0.6
## [49] parallel_4.2.1       listenv_0.9.0        crayon_1.5.2
## [52] splines_4.2.1        circlize_0.4.15      hms_1.1.3
## [55] knitr_1.43           pillar_1.9.0         boot_1.3-28.1
## [58] rjson_0.2.21         future.apply_1.11.0  reshape2_1.4.4
## [61] codetools_0.2-19     stats4_4.2.1         glue_1.6.2
## [64] evaluate_0.21        data.table_1.14.6    png_0.1-8
## [67] vctrs_0.6.2          tzdb_0.4.0           foreach_1.5.2
## [70] gtable_0.3.3         clue_0.3-63          future_1.33.0
## [73] xfun_0.39            gower_1.0.1          prodlim_2023.03.31
## [76] e1071_1.7-13         class_7.3-22         survival_3.5-5
## [79] timeDate_4022.108    iterators_1.0.14     IRanges_2.32.0
## [82] hardhat_1.3.0        cluster_2.1.4        lava_1.7.2.1
## [85] timechange_0.2.0     globals_0.16.2       ipred_0.9-14
```